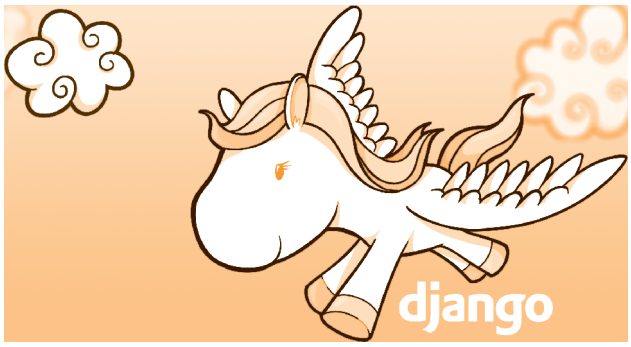
Choix d'un framework - Django

Gregory.Favre@epfl.ch
EPFL – Domaine IT, développeur Python pour le projet Infoscience



The advent of blogs, CMSs and RSSs have forever changed our manner to build the Web. Complex applications have gradually replaced HTML pages improved with CGI scripts. The high cost of development of very similar systems has made possible the ermergence of Web frameworks. Faced with the abondance of tools we have chosen Django.

L'arrivée des blogs, des CMS et des réseaux sociaux a définitivement changé notre manière de fabriquer le Web. Des applications complexes ont progressivement remplacé les pages HTML enrichies de scripts CGI. Le coût de développement important de systèmes au final très similaires a permis l'émergence des frameworks Web. Parmi la pléthore de ces outils, nous avons opté pour Django.

Le développement d'une application Web n'est pas un processus simple. Après avoir énuméré et décrit toutes les fonctions souhaitées, et avant même de pouvoir s'attaquer à toute forme de logique métier, il reste à:

- mettre en place une base de données;
- implémenter les tables et leurs relations;
- transformer tout ceci en HTML;
- définir les URL et comment elles se retrouveront associées aux données;
- gérer les utilisateurs et leurs droits d'accès;
- ...

C'est à ce stade qu'intervient le framework Web. Inutile de réinventer la roue, d'autres l'ont déjà fait, et – il faut l'admettre – considérablement mieux que nous.

Django - The Web framework for perfectionists with deadlines

Django est un framework *open source* basé sur python. Ses créateurs ont choisi de fournir un outil intégré, plutôt qu'un assemblage de projets préexistants (par opposition à son concurrent TurboGears).

Gestion des données

La création d'une application Web suppose généralement l'interaction avec une base de données. La connexion à une base ainsi que l'écriture de requêtes sont des tâches relativement simples, mais qui présentent des risques:

- types de données incompatibles entre le langage de programmation et le SGBD;
- syntaxe SQL incompatible entre deux SGBD différents;
- problèmes de sécurité (injection SQL).

Afin de pallier cela, Django propose un ORM, qui masque la complexité de la base de données au profit de simples classes python appelées modèles. Il devient ainsi aisé de développer dans un environnement léger comme SQLite et de passer en production sur un moteur plus solide comme PostgreSQL. Par ailleurs, le système s'assure lui-même de la conversion des données vers la base, déjouant au passage les actions malveillantes.
L'autre force de la gestion des modèles de données avec Django est la génération automatique des formulaires de recherche, d'ajout ou d'édition.

Génération d'HTML

Le développeur a souvent la mauvaise idée de mélanger l'HTML servant à l'interface avec la logique métier de son application, à plus forte raison s'il a été précédemment exposé à PHP. Le moindre changement de design impose alors d'aller modifier du code, avec tous les dangers que cela peut représenter.
Django propose un système de *templating*, permettant de séparer proprement le code HTML de la logique métier. Un graphiste peut ainsi facilement travailler sur l'affichage indépendamment des travaux réalisés par les développeurs. En outre, la structure proposée par Django est extrêmement souple et permet de facilement morceler une page en blocs fonctionnels séparés.

Gestion des URL

La différence de lisibilité entre l'adresse `monsite.com/blog/vive-python` et `monsite.com/blog.php?id=234` est flagrante. Posséder des adresses parlantes est non seulement un atout d'accessibilité, mais facilite en outre le travail des moteurs de recherche.
Django propose un système de gestion des URL efficace et simple, basé sur des listes d'expressions régulières. Ce système permet de découpler complètement les adresses du code.

Internationalisation

La traduction des applications Web est une évidence dans un pays quadrilingue. Django propose un système d'internationalisation facile à mettre en place, que ce soit dans les templates HTML ou dans le code d'une fonction métier. Les outils proposés savent scanner les fichiers et rassembler les phrases à traduire dans un seul fichier par langue.

Gestion des utilisateurs

Lorsque l'on doit gérer des transactions, garder un état cohérent tout au long du processus est indispensable. Qui plus est, on attend généralement d'une application de proposer un moyen d'identification, ainsi qu'une gestion des droits. La complexité d'un tel système rend son développement fastidieux. Django propose une gestion intégrée des utilisateurs. Ce système est extrêmement facile à personnaliser de manière à offrir, par exemple, une authentification via Shibboleth.

Conclusion

Django permet de remplir toutes nos attentes vis-à-vis d'un système. Il permet de s'économiser du temps de développement et par la même de focaliser sur ce qui compte vraiment: la logique métier. À titre d'exemple, un système de blog simple avec interface d'administration tient en moins de 60 lignes de code! ■

GLOSSAIRE

ORM *(Object-Relational Mapper)*: technique de programmation faisant le lien entre le monde de la base de données et le monde de la programmation objet.

SGBD: Système de Gestion de Bases de Données. Ensemble de logiciels servant à la manipulation de bases de données.

Shibboleth: mécanisme permettant de déléguer l'authentification d'un utilisateur à son établissement d'origine.

La couleur 2010

Analyse

FlashiPhone –

Francois.Roulet@epfl.ch
EPFL - Domaine IT, heureux utilisateur de iPhone

QR Code Recognition

Reconnaissance du QR Code

Bien que le **QR Code** ne soit pas nouveau, l'actualité régionale l'a récemment fait ressurgir, depuis son usage pour la diffusion des horaires en temps réel par les **T-L** (Transports publics lausannois). En effet, depuis le mois de décembre 2009, les T-L ont apposé les QR Code de l'horaire instantané du métro m1 à chaque arrêt. Dès lors, n'importe quel téléphone mobile doté d'un appareil photo, pourra lire et décoder le QR Code, qui vous redirigera aussitôt sur la page Web affichant l'horaire en temps réel de la ligne et de l'arrêt correspondants.

Le **QR Code**, acronyme de *Quick Response Code*, est une évolution du code matriciel, ou code bi-dimensionnel, dont la lecture est notablement accélérée. Il cumule les qualités individuelles de ses prédécesseurs, à savoir une grande capacité de données sur une faible surface imprimée. Son domaine d'application est plus vaste que le simple référencement de produits, et s'étend notamment à la transmission d'adresses, URL et Vcard.

Nous sommes tous familiers des codes barres unidimensionnels qui étaient déjà largement répandus depuis trois décennies aux USA pour l'étiquetage de produits commerciaux, avant de traverser l'Atlantique plus récemment et devenir le **EAN** *(European Article Number)*. Il s'est notamment imposé dans l'édition, avec le célèbre code **ISBN** *(International Standard Book Number)*, ou encore **ISSN** *(International Standard Serial Number)* pour les publications en série, tel notre journal – voir le code barre en dernière page. Ces codes unidimensionnels, se reconnaissent aisément à leur alignement de barres noires rectilignes sur fond blanc, d'épaisseur et d'espacement variables.

Ensuite, sont apparus les codes bidimensionnels **DataMatrix**, que vous observez sur le courrier postal, dont la densité d'information véhiculée, de par sa nature quadratique, est sensiblement supérieure, pour dépasser 3000 chiffres. D'autres codes plus imposants sont les **Stacked Bar**, codes à balayage, qui ornent l'entête des déclarations fiscales vaudoises. Ces évolutions ont pour but de mettre à disposition une charge utile bien supérieure, qui n'est plus limitée à quelques chiffres. Selon la taille et le niveau